W126–W131 *Nucleic Acids Research, 2007, Vol. 35, Web Server issue*
doi:10.1093/nar/gkm219

OPTIMIZER: a web server for optimizing the codon usage of DNA sequences

Pere Puigbò[1], Eduard Guzmán[1,2], Antoni Romeu[1] and Santiago Garcia-Vallvé[1,]*

[1]Evolutionary Genomics Group, Biochemistry and Biotechnology Department, Faculty of Chemistry, Rovira i Virgili University (URV), c/Marcel·li Domingo, s/n. Campus Sescelades, 43007 Tarragona, Spain and [2]Institut Català de la Salut, Àrea Bàsica de Salut, Tarragona 2, Spain

Received January 29, 2007; Revised March 22, 2007; Accepted March 28, 2007

ABSTRACT

***OPTIMIZER* is an on-line application that optimizes the codon usage of a gene to increase its expression level. Three methods of optimization are available: the 'one amino acid–one codon' method, a guided random method based on a Monte Carlo algorithm, and a new method designed to maximize the optimization with the fewest changes in the query sequence. One of the main features of *OPTIMIZER* is that it makes it possible to optimize a DNA sequence using pre-computed codon usage tables from a predicted group of highly expressed genes from more than 150 prokaryotic species under strong translational selection. These groups of highly expressed genes have been predicted using a new iterative algorithm. In addition, users can use, as a reference set, a pre-computed table containing the mean codon usage of ribosomal protein genes and, as a novelty, the tRNA gene-copy numbers. *OPTIMIZER* is accessible free of charge at http://genomes.urv.es/OPTIMIZER.**

INTRODUCTION

Gene expression levels depend on many factors, such as promoter sequences and regulatory elements. One of the most important factors is the adaptation of the codon usage of the transcript gene to the typical codon usage of the host (1). Therefore, highly expressed genes in prokaryotic genomes under translational selection have a pronounced codon usage bias. This is because they use a small subset of codons that are recognized by the most abundant tRNA species (2). The force that modulates this codon adaptation is called translational selection and its strength is important in fast-growing bacteria (3,4). If a gene contains codons that are rarely used by the host, its expression level will not be maximal. This may be one of the limitations of heterologous protein expression (5) and the development of DNA vaccines (6). A high number of synthetic genes have been re-designed to increase their expression level. The Synthetic Gene Database (SGDB) (7) contains information from more than 200 published experiments on synthetic genes. In the design process of a nucleic acid sequence that will be inserted into a new host to express a certain protein in large amounts, codon usage optimization is usually one of the first steps (5). Codon usage optimization basically involves altering the rare codons in the target gene so that they more closely reflect the codon usage of the host without modifying the amino acid sequence of the encoded protein (5). The information usually used for the optimization process is therefore the DNA or protein sequence to be optimized and a codon usage table (which we call the reference set) of the host.

Here we present a new web server, called *OPTIMIZER*, for codon usage optimization focused on the heterologous, or even homologous, gene expression in bacterial hosts. *OPTIMIZER* allows three optimization methods and uses several valuable, new reference sets. *OPTIMIZER* can therefore be used to optimize the expression level of a gene, to assess the adaptation of alien genes inserted into a genome (8), or to design new genes from protein sequences. The server is freely available at http://genomes.urv.es/OPTIMIZER. It has been running since July 2005 and it is updated twice a year with new features and reference sets.

PROGRAM OVERVIEW

Implementation and input data

OPTIMIZER is an on-line application and its methods are implemented in PHP (hypertext pre-processor) programming language. The pre-calculated reference tables are stored into a MySQL database. The data input and the selection of the server options have been organized in four steps. These steps are: (1) Input the sequence to be optimized. DNA or protein sequences can be used, although further steps are slightly different depending on whether a DNA or protein sequence has been input. (2) Input the reference set. Users can insert

*To whom correspondence should be addressed. Tel: +34 977558778; Fax: +34 977558232; Email: santi.garcia-vallve@urv.net a codon usage table in a variety of formats, including tables from the Codon Usage Database (9), or they can choose between 153 pre-computed codon usage tables for ribosomal protein genes or a group of highly expressed genes from prokaryotic genomes under translational selection. Users can also choose a reference set consisting of the tRNA gene-copy numbers. (3) Choose the genetic code. (4) Choose the method to be used in the optimization process. Depending on the type of sequence introduced (DNA or protein) and the reference set chosen, different optimization methods are available (see below for a description of the optimization methods).

Calculation of the reference sets

One of the main features of the *OPTIMIZER* server is that it contains a series of pre-computed reference sets that can be used in the optimization process. These reference sets can be a table containing the codon usage of the host (or the codon usage of a group of genes, such as the group of highly expressed genes) or, as a novelty, the number of tRNA gene copies predicted with the tRNA-scan software (10). The pre-computed reference sets available in the server are from more than 150 prokaryotic genomes that are under a strong translational selection. The codon usage reference tables available for these genomes contain the mean codon usage of genes that encode ribosomal proteins or a group of highly expressed genes. Although the optimization process can be carried out using the mean codon usage of the host organism as a reference set, if the aim of the optimization process is to increase the expression level of a gene, it is preferable to use the codon usage of a group of highly expressed genes. The mean codon usage of bacteria is highly influenced by mutational bias (i.e. their G + C content). The optimal codons (those most frequently used in highly expressed genes) are usually those that agree with the mutational bias (i.e. G- or C-ending codons for G + C-rich organisms). However, the optimal codons are not always in agreement with mutational bias. For example, in the amino acids that are coded by only two synonymous codons ending in C or T, the C-ending codon is usually preferred, independently of the mutational bias (3). Therefore, using the mean codon usage of a genome may cause the wrong choice of optimal codons.

A new feature of the *OPTIMIZER* server is that it can use tRNA gene-copy numbers as a reference set for the optimization process. If the codon usage bias of highly expressed genes is caused by differences in tRNA gene-copy numbers, why not use this information for the optimization process? At present, information about tRNA gene-copy numbers is used in the *OPTIMIZER* server only with the 'one amino acid–one codon' optimization method (for a complete description of the methods available, see the 'Optimization methods' section below).

Evaluation of which bacterial genomes are under translational selection. Not all prokaryotic species are under translational selection (4,11). It would be pointless to optimize the codon usage of a gene in order to increase its expression level in a species such as *Helicobacter pylori*, which is not under translational selection (i.e. in which the highly expressed genes do not have a different pattern of codon usage from the other genes of their genome) (12). Traditionally, correspondence analysis of the relative synonymous codon usage of all genes from a genome has been used to detect whether a genome is under translational selection (13). In genomes under translational selection, the ribosomal protein genes and other highly expressed genes form a cluster in the correspondence analysis plot, which confirms that highly expressed genes have a different codon usage from the other genes of a genome. This is the method we have used to detect which bacterial species are under translational selection. For each bacterial complete genome available, we made a correspondence analysis using the Relative Synonymous Codon Usage (RSCU) values of all the genes of a genome. To automate the analysis of the correspondence plots, we analyzed the position of the ribosomal protein genes (expected to be highly expressed genes) along the four principal axes obtained in the correspondence analysis. If a genome is under translational selection, ribosomal proteins and other highly expressed genes will show a codon usage bias and they will form a cluster in the correspondence plot. To make the prediction of translational selection, we checked whether the mean position of the ribosomal protein genes along any of the four principal axes was significantly different (evaluated with a *t*-test) from the mean position of the other genes of their genome. To check our predictions, we also visually inspected the correspondence plots (correspondence analysis plots are available from the homepage of the server) and analyzed the metabolic function of the predicted highly expressed genes obtained. Analysis of 334 prokaryotic genomes revealed that 153 genomes (the total number of different species and genera was 108 and 63, respectively) were under a strong translational selection. These genomes were then used to calculate the pre-computed reference sets.

Prediction of highly expressed genes. The predicted highly expressed genes were obtained using an iterative algorithm that we have developed. This algorithm uses the group of genes that encode ribosomal proteins as a seed and, through a series of iterations, define a group of putative highly expressed genes. This algorithm works as follows:

(i) Using the functional annotation, gene names or COG families, genes that encode ribosomal proteins are detected. Using the codon usage of these genes as a reference set, the Codon Adaptation Index (CAI), (14), at this stage namely CAIrp (15), is calculated for each gene of a genome.

(ii) Using now the group of genes with the highest CAI values as a reference set, the CAI for all genes is recalculated. This process is repeated until a homogeneous group is reached, i.e. when the group of genes with the highest CAI values in one iteration is the same as the group in the next iteration.

To provide further support for our predictions, we analyzed the metabolic functions of the putative highly expressed genes. As expected, ribosomal proteins and other expected highly expressed genes (16) were found in the final group of predicted highly expressed genes. To check our algorithm, we also analyzed species not under translational selection. With these species, either the algorithm never ended or the final group of genes had a high codon usage bias but was not related to their expression level. In this situation, neither ribosomal protein genes nor genes expected to have a high expression were included in the final group of genes with a codon usage bias. Our method is similar to the one developed by Carbone and co-workers (17). However, these authors used all the genes of an organism as the initial reference set, whereas we used ribosomal protein genes.

Optimization methods

The *OPTIMIZER* server provides three methods for optimizing the codon usage of the query sequence. In the first method, the 'one amino acid–one codon' method, all the codons that encode the same amino acid are substituted by the most commonly used synonymous codon in the reference set. However, this approach has several drawbacks: for example, translational errors may be made due to an imbalanced tRNA pool and it is impossible to avoid repetitive elements or cleavage sites of restriction enzymes (5,18). To overcome these drawbacks, a second method, which we call the 'guided random' method, can be used. This method consists of a Monte Carlo algorithm that selects codons at random based on the frequencies of use of each codon in the reference set. The third method, which we call the 'customized one amino acid–one codon' method, is an intermediate method in which users choose how many of the 59 codons (if the standard genetic code has been selected) will be optimized with the 'one amino acid–one codon' approach. 'Rare codons' (i.e. the least used codons in the reference set) are the first codons changed with this approach. The aim of this third method is to maximize the optimization by making the fewest changes in the query sequence.

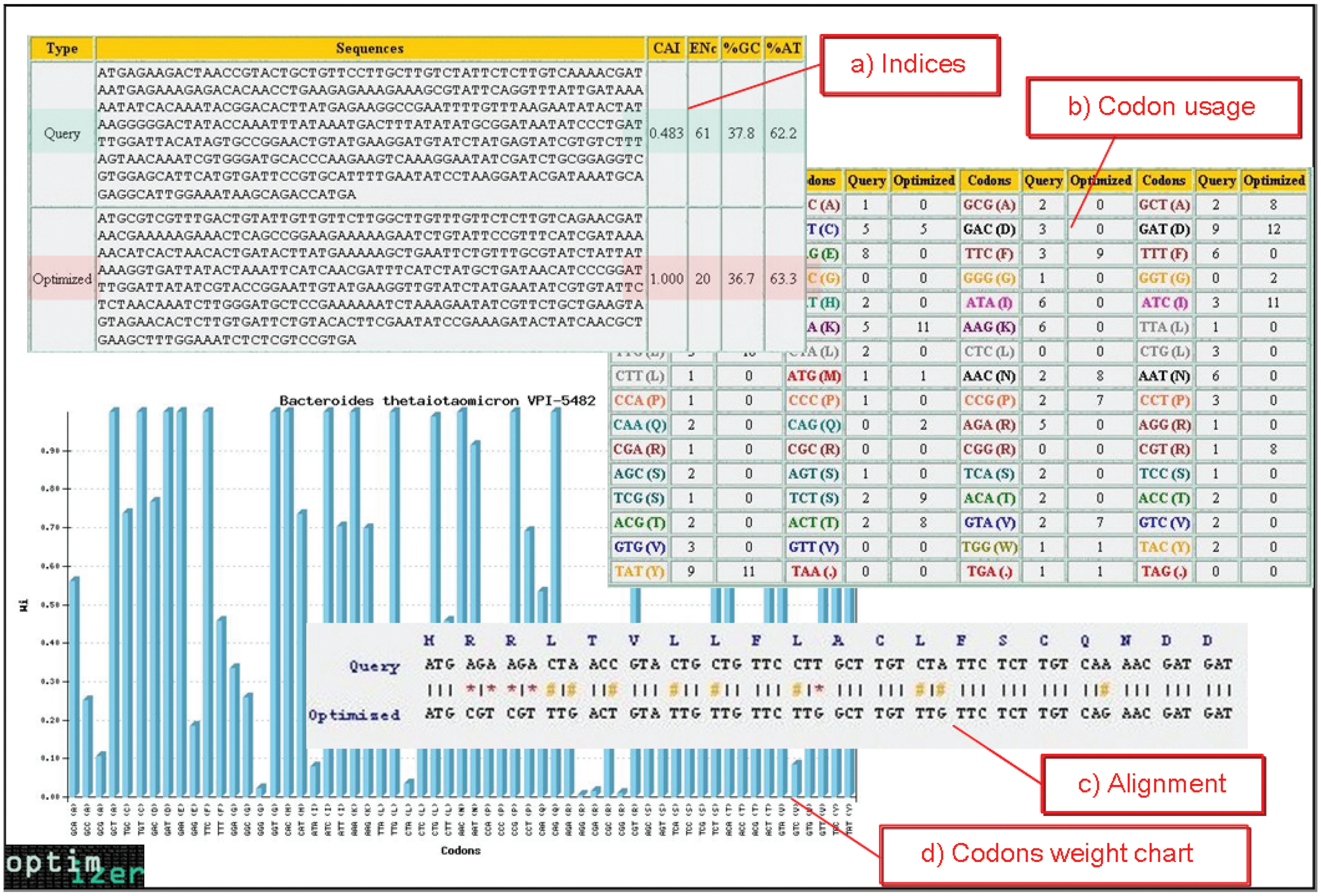
If the input sequence is a protein, it can be back-translated to DNA using the 'one amino acid–one codon' or the 'guided random' approach. If the 'one amino acid–one codon' approach is selected, the protein sequence can be back-translated to DNA using codons with the highest G + C or A + T content, or codons defined by Archetti (19) that minimize mutation errors.

Outputs

Two indices, CAI and ENc (effective number of codons), are used to measure the optimization process. CAI measures the similarity between the codon usage of a gene and the codon usage of a reference group of genes (14). Its values range from 0 (when the codon usage of a sequence and that of the reference set are very different) to 1 (when both codon usages are the same). This index is the most effective of all codon bias measures for predicting gene expression levels (12,20). The second index is ENc, which is a measure of codon usage bias (21). Its values range from 20 (if only one codon per amino acid is used) to 61 (if all synonymous codons are used equally). Because highly expressed genes usually use the minimal subset of codons that are recognized by the most abundant tRNA species, their ENc values are expected to be low. Figure 1 shows some of the outputs provided by the optimization of a DNA sequence: for example, the query and optimized sequences and an alignment between them, a chart of the relative frequencies of each codon of the reference set and a codon usage table of the query and optimized sequences. In addition, the *OPTIMIZER* server has options for viewing or avoiding the cleavage sites of the selected restriction enzymes (22) and for splitting the optimized sequence into several overlapping oligonucleotides for the construction of a synthetic gene.

Comparison with other servers and programs

Table 1 shows a comparison of several public web servers and stand-alone applications that allow some kind of codon optimization. '*GeneDesign*' (23), '*Synthetic Gene Designer*' (24) and '*Gene Designer*' (18) are packages that provide a platform for synthetic gene design, including a codon optimization step. Other programs, such as *DNAWorks* (25) and GeMS (26), focus more on the process of oligonucleotide design for synthetic gene construction. The stand-alone application *INCA* provides an array of features, including now codon optimization, which are useful for analyzing synonymous codon usage in whole genomes (27). *JCAT* (28), '*Codon optimizer*' (29), *UpGene* (30) and the server presented here focus on the codon optimization process. Although each server and application has its own features, all of them have several features in common. Most offer several options for the input of the codon usage reference set. One of these options is the possibility of using the tables from the Codon Usage database (9). Usually, a limited number of pre-computed tables of codon usage are available to be used as a reference set in the optimization process. In addition, not all of the available pre-computed reference sets correspond to a group of highly expressed genes (the proper reference set needed to optimize for increasing gene expression level). Though most of the programs and servers use a group of highly expressed genes from *E. coli* as a pre-computed reference set, only the '*Synthetic Gene Designer*' and '*GeneDesign*' servers provide a pre-computed group of highly expressed genes for 11 and 4 organisms, respectively. The exception is the *JCAT* web server, which offers pre-computed tables of predicted highly expressed genes from more than 200 bacterial species. However, this server uses the method of Carbone *et al.* (17) to predict a group of genes with a biased codon usage. These groups of genes do not always correspond to a group of highly expressed genes because not all bacterial species are under translational selection (11,17). The high number of pre-computed codon usage tables from bacteria and archaea that are



Type	Sequences	CAI	ENc	%GC	%AT
Query	ATGAGAAGACTAACCGTACTGCTGTTCCTTGCTTGTCTATTCTCTTGTCAAAACGAT AATGAGAAAGAGACACAACCTGAAGAGAAAGAAAGCGTATTCAGGTTTATTGATAAA AATATCACAAATACGGACACTTATGAGAAGGCCGAATTTTGTTTAAGAATATACTAT AAGGGGGACTATACCAAATTTATAAATGACTTTATATATGCGGATAATATCCCTGAT TTGGATTACATAGTGCCGGAACTGTATGAAGGATGTATCTATGAGTATCGTGTCTTT AGTAACAAATCGTGGGATGCACCCAAGAAGTCAAAGGAATATCGATCTGCGGAGGTC GTGGAGCATTCATGTGATTCCGTGCATTTTGAATATCCTAAGGATACGATAAATGCA GAGGCATTGGAAATAAGCAGACCATGA	0.483	61	37.8	62.2
Optimized	ATGCGTCGTTTGACTGTATTGTTGTTCTTGGCTTGTTTGTTCTCTTGTCAGAACGAT AACGAAAAAGAAACTCAGCCGGAAGAAAAAGAATCTGTATTCCGTTTCATCGATAAA AACATCACTAACACTGATACTTATGAAAAAGCTGAATTCTGTTTGCGTATCTATTAT AAAGGTGATTATACTAAATTCATCAACGATTTCATCTATGCTGATAACATCCCGGAT TTGGATTATATCGTACCGGAATTGTATGAAGGTTGTATCTATGAATATCGTGTATTC TCTAACAAATCTTGGGATGCTCCGAAAAAATCTAAAGAATATCGTTCTGCTGAAGTA GTAGAACACTCTTGTGATTCTGTACACTTCGAATATCCGAAAGATACTATCAACGCT GAAGCTTTGGAAATCTCTCGTCCGTGA	1.000	20	36.7	63.3

			dons	Query	Optimized	Codons	Query	Optimized	Codons	Query	Optimized
			C (A)	1	0	GCG (A)	2	0	GCT (A)	2	8
			T (C)	5	5	GAC (D)	3	0	GAT (D)	9	12
			G (E)	8	0	TTC (F)	3	9	TTT (F)	6	0
			C (G)	0	0	GGG (G)	1	0	GGT (G)	0	2
			T (H)	2	0	ATA (I)	6	0	ATC (I)	3	11
			A (K)	5	11	AAG (K)	6	0	TTA (L)	1	0
TTG (L)	3	10	CTA (L)	2	0	CTC (L)	0	0	CTG (L)	3	0
CTT (L)	1	0	ATG (M)	1	1	AAC (N)	2	8	AAT (N)	6	0
CCA (P)	1	0	CCC (P)	1	0	CCG (P)	2	7	CCT (P)	3	0
CAA (Q)	2	0	CAG (Q)	0	2	AGA (R)	5	0	AGG (R)	1	0
CGA (R)	1	0	CGC (R)	0	0	CGG (R)	0	0	CGT (R)	1	8
AGC (S)	2	0	AGT (S)	1	0	TCA (S)	2	0	TCC (S)	1	0
TCG (S)	1	0	TCT (S)	2	9	ACA (T)	2	0	ACC (T)	2	0
ACG (T)	2	0	ACT (T)	2	8	GTA (V)	2	7	GTC (V)	2	0
GTG (V)	3	0	GTT (V)	0	0	TGG (W)	1	1	TAC (Y)	2	0
TAT (Y)	9	11	TAA (.)	0	0	TGA (.)	1	1	TAG (.)	0	0

```
                 M   R   R   L   T   V   L   L   F   L   A   C   L   F   S   C   Q   N   D   D
     Query      ATG AGA AGA CTA ACC GTA CTG CTG TTC CTT GCT TGT CTA TTC TCT TGT CAA AAC GAT GAT
                ||| *|* *|* #|# ||# ||| #|| #|| ||| #|* ||| ||| #|# ||| ||| ||| ||# ||| ||| |||
Optimized       ATG CGT CGT TTG ACT GTA TTG TTG TTC TTG GCT TGT TTG TTC TCT TGT CAG AAC GAT GAT
```

Figure 1. Outputs provided from the optimization of a DNA sequence: (**a**) the optimized and query sequences and the indices (CAI, ENc and%G + C) for evaluating the optimization process, (**b**) codon usage tables of the query and optimized sequences, (**c**) query and optimized sequence alignment to show changes in nucleotides (transitions or transversions) and (**d**) graphical view of the codon weight chart.

not under translational selection available in *JCAT* therefore creates some confusion. The *OPTIMIZER* server presented here provides the most pre-computed codon usage tables for use as a reference set. The *OPTIMIZER* server provides pre-computed tables for more than 150 prokaryotic genomes that are under strong translational selection. In addition, two groups of genes are available in each reference set: a group of highly expressed genes predicted using a new prediction algorithm and the group of ribosomal protein genes. *OPTIMIZER* is the only server or stand-alone application that introduces a new kind of reference set such as information about the number of copies of tRNA genes for all the species included in the server. With regard to the methods for codon usage optimization available in each server or program, the first programs developed used only the 'one amino acid–one codon' approach. More recent programs and servers now include further methods to create some codon usage variability. This variability reflects the codon usage variability of natural highly expressed genes and enables additional criteria to be introduced (such as the avoidance of restriction sites) in the optimization process. The *OPTIMIZER* server presented here provides three methods of codon optimization: a complete optimization of all codons, an optimization based on the relative codon usage frequencies of the reference set that uses a Monte Carlo approach (similar to methods from other programs and servers) and a novel approach designed to maximize the optimization with the minimum changes between the query and optimized sequences. Finally, note that only the '*Synthetic Gene Designer*,' *INCA* and *OPTIMIZER* allow users to choose a non-standard genetic code.

CONCLUSIONS

OPTIMIZER is a new codon optimization web server focused on maximizing the gene expression level through the optimization of codon usage. It has unique features, such as a novel definition of a group of highly expressed genes from more than 150 prokaryotic species under translational selection, and the possibility of using information on tRNA gene-copy numbers in the optimization process. *OPTIMIZER* provides several pre-computed tables to specify a reference set and combines

Table 1. Comparison of *OPTIMIZER* with other similar freely available web servers and softwares

Name	Methods	Genetic code	Reference set	Reference
Web servers				
OPTIMIZER	– One amino acid–one codon – Guided Random (Monte Carlo algorithm)[b] – Customized one amino acid–one codon	Multiple	– HEG from >150 bacterial genomes under TS – RPG – tGCN – Codon usage database – Defined by users	This article
JCAT	– One amino acid–one codon	Standard	– HEG from >200 bacterial genomes – Defined by users	28
Synthetic Gene Designer (SGD)	– One amino acid–one codon[a] – Selective optimization[a] – Probabilistic optimization[a,b]	Multiple	– HEG from six bacterial genomes – Codon usage database – Defined by users	24
DNAWorks	– Use of the two highest frequency codons – Random	Standard	– HEG from *E. coli* – Codon usage tables for 10 species – Codon usage database – Defined by users	25
GeneDesign	– One amino acid–one codon – The next most optimal algorithm – The most different algorithm – Random	Standard	– HEG from four species – Defined by users	23
Stand-alone applications				
Gene Designer	– One amino acid–one codon – Monte Carlo algorithm[b]	Standard	– HEG from *E. coli* – Codon usage tables for 25 species – Codon usage database – Defined by users	18
Codon optimizer	– One amino acid–one codon	Standard	– HEG for several bacterial species – Defined by users	29
INCA 2.1	– One amino acid–one codon	Multiple	– Mean codon usage of a whole genome or selection of any group of genes	27
UPGene	– One amino acid–one codon	Standard	– Eukaryotic, bacteria, yeast, plant and worm predefined codon usage frequency tables – Defined by users	30
GeMS	– Monte Carlo algorithm[b]	Standard	– Codon usage database – Defined by users	26

Abbreviations used: HEG, codon usage of predicted highly expressed genes; RPG, codon usage of ribosomal protein genes; tGCN, tRNA gene-copy number; TS, translational selection.

[a] It uses an 'optimality factor,' defined as a scaling factor, to control the optimality of codon usage. Higher values of this factor mean low CAI values and less optimized and more random codon usage.

[b] These methods are essentially the same. They use the relative codon usage frequencies of the reference set as the relative probability that each codon will be used in the optimization process.

three different methods of codon optimization. The *OPTIMIZER* server can be used to optimize the expression level of a gene in heterologous gene expression or to design new genes that confer new metabolic capabilities in a given species.

ACKNOWLEDGEMENTS

This work has been financed by project BIO2003-07672 of the Spanish Ministry of Science and Technology. We thank Kevin Costello and John Bates of the Language Service of the Rovira i Virgili University for their help in writing the manuscript and two anonymous referees for their helpful comments. Funding to pay the Open Access publication charges for this article was provided by project BIO2003-07672 of the Spanish Ministry of Science and Technology.

Conflict of interest statement. None declared.

REFERENCES

1. Lithwick,G. and Margalit,H. (2003) Hierarchy of sequence-dependent features associated with prokaryotic translation. *Genome Res.*, **13**, 2665–2673.
2. Ikemura,T. (1981) Correlation between the abundance of Escherichia coli transfer RNAs and the occurrence of the respective codons in its protein genes: a proposal for a synonymous codon choice that is optimal for the *E. coli* translational system. *J. Mol. Biol.*, **151**, 389–409.
3. Rocha,E.P. (2004) Codon usage bias from tRNA's point of view: redundancy, specialization, and efficient decoding for translation optimization. *Genome Res.*, **14**, 2279–2286.
4. Sharp,P.M., Bailes,E., Grocock,R.J., Peden,J.F. and Sockett,R.E. (2005) Variation in the strength of selected codon usage bias among bacteria. *Nucleic Acids Res.*, **33**, 1141–1153.
5. Gustafsson,C., Govindarajan,S. and Minshull,J. (2004) Codon bias and heterologous protein expression. *Trends Biotechnol.*, **22**, 346–353.
6. Ivory,C. and Chadee,K. (2004) DNA vaccines: designing strategies against parasitic infections. *Genet. Vaccines Ther.*, **2**, 17.
7. Wu,G., Zheng,Y., Qureshi,I., Zin,H.T., Beck,T., Bulka,B. and Freeland,S.J. (2007) SGDB: a database of synthetic genes

re-designed for optimizing protein over-expression. *Nucleic Acids Res.*, **35**, D76–D79.
8. Garcia-Vallve,S., Guzman,E., Montero,M.A. and Romeu,A. (2003) HGT-DB: a database of putative horizontally transferred genes in prokaryotic complete genomes. *Nucleic Acids Res.*, **31**, 187–189.
9. Nakamura,Y., Gojobori,T. and Ikemura,T. (2000) Codon usage tabulated from international DNA sequence databases: status for the year 2000. *Nucleic Acids Res.*, **28**, 292.
10. Lowe,T.M. and Eddy,S.R. (1997) tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Res.*, **25**, 955–964.
11. Willenbrock,H., Friis,C., Friis,A.S. and Ussery,D.W. (2006) An environmental signature for 323 microbial genomes based on codon adaptation indices. *Genome Biol.*, **7**, R114.
12. Henry,I. and Sharp,P.M. (2007) Predicting gene expression level from codon usage bias. *Mol. Biol. Evol.*, **24**, 10–12.
13. Perriere,G. and Thioulouse,J. (2002) Use and misuse of correspondence analysis in codon usage studies. *Nucleic Acids Res.*, **30**, 4548–4555.
14. Sharp,P.M. and Li,W.H. (1987) The codon adaptation index – a measure of directional synonymous codon usage bias, and its potential applications. *Nucleic Acids Res.*, **15**, 1281–1295.
15. Nakamura,Y. and Tabata,S. (1997) Codon-anticodon assignment and detection of codon usage trends in seven microbial genomes. *Microb. Comp. Genomics*, **2**, 299–312.
16. Karlin,S. and Mrazek,J. (2000) Predicted highly expressed genes of diverse prokaryotic genomes. *J. Bacteriol.*, **182**, 5238–5250.
17. Carbone,A., Zinovyev,A. and Kepes,F. (2003) Codon adaptation index as a measure of dominating codon bias. *Bioinformatics*, **19**, 2005–2015.
18. Villalobos,A., Ness,J.E., Gustafsson,C., Minshull,J. and Govindarajan,S. (2006) Gene designer: a synthetic biology tool for constructing artificial DNA segments. *BMC Bioinformatics*, **7**, 285.
19. Archetti,M. (2004) Selection on codon usage for error minimization at the protein level. *J. Mol. Evol.*, **59**, 400–415.
20. Goetz,R.M. and Fuglsang,A. (2005) Correlation of codon bias measures with mRNA levels: analysis of transcriptome data from Escherichia coli. *Biochem. Biophys. Res. Commun.*, **327**, 4–7.
21. Wright,F. (1990) The 'effective number of codons' used in a gene. *Gene*, **87**, 23–29.
22. Roberts,R.J., Vincze,T., Posfai,J. and Macelis,D. (2005) REBASE – restriction enzymes and DNA methyltransferases. *Nucleic Acids Res.*, **33**, D230–D232.
23. Richardson,S.M., Wheelan,S.J., Yarrington,R.M. and Boeke,J.D. (2006) GeneDesign: rapid, automated design of multikilobase synthetic genes. *Genome Res.*, **16**, 550–556.
24. Wu,G., Bashir-Bello,N. and Freeland,S.J. (2006) The synthetic gene designer: a flexible web platform to explore sequence manipulation for heterologous expression. *Protein Expr. Purif.*, **47**, 441–445.
25. Hoover,D.M. and Lubkowski,J. (2002) DNAWorks: an automated method for designing oligonucleotides for PCR-based gene synthesis. *Nucleic Acids Res.*, **30**, e43.
26. Jayaraj,S., Reid,R. and Santi,D.V. (2005) GeMS: An advanced software package for designing synthetic genes. *Nucleic Acids Res.*, **33**, 3011–3016.
27. Supek,F. and Vlahovicek,K. (2004) INCA: Synonymous codon usage analysis and clustering by means of self-organizing map. *Bioinformatics*, **20**, 2329–2330.
28. Grote,A., Hiller,K., Scheer,M., Munch,R., Nortemann,B., Hempel,D.C. and Jahn,D. (2005) JCat: a novel tool to adapt codon usage of a target gene to its potential expression host. *Nucleic Acids Res.*, **33**, W526–W531.
29. Fuglsang,A. (2003) Codon optimizer: a freeware tool for codon optimization. *Protein Expr. Purif.*, **31**, 247–249.
30. Gao,W., Rzewski,A., Sun,H., Robbins,P.D. and Gambotto,A. (2004) UpGene: Application of a web-based DNA codon optimization algorithm. *Biotechnol. Prog.*, **20**, 443–448.